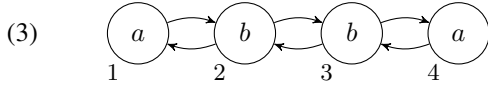


The string model is represented graphically in (3) where P_a and P_b are indicated by labels on the nodes, and p and s are indicated by arrows.



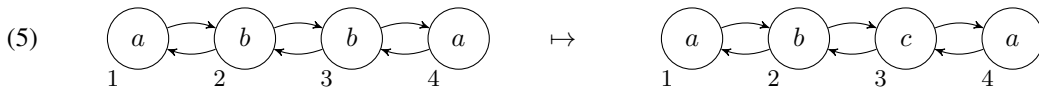
The relations and functions in this model comprise a *signature* $\mathcal{S} = \{P_a, P_b, p, s\}$ which we may use to represent any string of a s and b s: $aabb$, $ababba$, ba , etc.

Additionally, the relations and functions induce a *logical language* with which we make statements about the models themselves. This language is a set of atomic formulae based on the relations/functions in the signature. Formulae take variables (x , y , etc.) that are interpreted as elements in some model. For example, the atomic formula $P_a(x)$ will be true in a model for some assignment of x to a domain element which is in that relation in the model. In the model for $abba$, $P_a(x)$ is true when x is assigned to element 1 or 4, but is false when x is assigned to element 2 or 3. We recurse over atomic predicates using Boolean connectives (negation \neg , disjunction \vee , conjunction \wedge , and implication \rightarrow) to define a full set of formulas. For example, the formula $P_a(x) \vee P_b(x)$ will be true of assignments of x to any of $\{1,2,3,4\}$ in the model of $abba$.

We may also use this logical language to define mappings from an input structure to an output structure. These *logical transductions* (Courcelle, 1994; Engelfriet & Hooeboom, 2001) are defined via a set of logical formulas, one for each relation and function in the *output* signature. They are interpreted with respect to the logical language of the *input* signature. For example, consider the rule $b \rightarrow c / _a$. Assume that the application of the rule is non-iterative, such that when applied to $abba$ it yields $abca$. A *logical* definition of this rule is a transduction defined over an output signature $\mathcal{S}_o = \{P'_a, P'_b, P'_c, p', s'\}$. To model this rule applied to $abba$ above, the transduction is interpreted with respect to its signature $\mathcal{S}_i = \{P_a, P_b, p, s\}$, that is, the input signature. We define the transduction below.

$$(4) \quad \begin{aligned} P'_a(x) &\stackrel{d}{=} P_a(x) & p'(x) &\stackrel{d}{=} p(x) \\ P'_b(x) &\stackrel{d}{=} P_b(x) \wedge \neg P_a(s(x)) & s'(x) &\stackrel{d}{=} s(x) \\ P'_c(x) &\stackrel{d}{=} P_b(x) \wedge P_a(s(x)) \end{aligned}$$

Applied to the model for $abba$ in (2), the output mapping is:



$P'_a(x)$ in the output is defined identically to the input, such that any domain element labeled a in the input will also be marked a in the output (nodes 1 and 4). The definition of $P'_b(x)$ states that any input element marked b which is not immediately followed by an a will be marked b in the output (node 2). The output definition $P'_c(x)$ states that any input element marked b in the input and which is immediately followed by an a is labeled c . Note that linear order on the output ($p'(x)$ and $s'(x)$) does not change.

The reader can confirm that the transduction in (4) will change any input string of a s and b s to an identical string, except for any b that was followed by an a in the input will be changed to a c . Thus, this transduction represents a logical definition of the rewrite rule $b \rightarrow c / _a$, as the two describe exactly the same function.

2.2 Least-Fixed Point Operators The logical definitions above are *quantifier-free* (QF); that is, they make no use of the logical quantifiers \exists and \forall to introduce additional variables. A QF logic is inherently *local* because, for example, a QF formula $\phi(x)$ with a free variable x cannot refer to positions arbitrarily far away from x . In fact, QF logical transductions correspond to *input strictly local functions* (Chandlee & Lindell, forthcoming), which are string functions whose outputs are computed deterministically by considering the local window around each position in the input string (Chandlee, 2014). A great many phonological processes are input strictly local (Chandlee & Heinz, 2018).

However, not *all* phonological processes are input strictly local and thus describable by QF transductions. For example, consider the process described by iterative application of the rule $a \rightarrow b / b _$. Consider the two mappings below.

- (6) a. $baaaa \mapsto bbbbb$
 b. $aaaaa \mapsto aaaaa$

The final *as* in the two inputs have different outputs: in (6a) it is *b*, whereas in (6b) it is *a*. This is not input strictly local because the original trigger—the input *b*—can be arbitrarily far away in the *input* from any of its eventual targets. In logical terms, we cannot define an output formula $P'_b(x)$ for this function in QF because no finite application $p(p(\dots p(x)\dots))$ of the predecessor function to x is going to identify all possible input positions that the original triggering *b* could occupy.

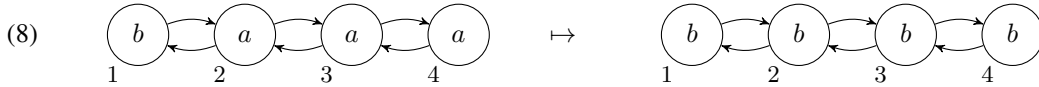
Intuitively, however, this is *output*-local because the trigger for spread is adjacent in the output; that is, the final *a* in (6a) becomes a *b* because the preceding *output* symbol is a *b*, whereas in (6b) the final *a* stays an *a* because the preceding output symbol is an *a*. We can capture such iterative, output-based locality with *least-fixed point* logics (Libkin, 2004).

A least-fixed point logic allows for simple recursive definitions of predicates. We refrain from a full exploration of the formalism here, and instead illustrate how recursive logical formulae can capture output-based locality. Instead of the full least-fixed point formalism of, e.g., Libkin (2004), we use *implicit definitions* of predicates (Rogers, 1997).

An implicit definition is the definition of a predicate that recursively refers to that predicate itself. For example, for the spreading rule in (6) we can define a predicate $P'_b(x)$ for output *bs* recursively as in (7).

$$(7) \quad P'_b(x) \stackrel{d}{=} P_b(x) \vee P'_b(p(x))$$

This formula is interpreted as follows: $P'_b(x)$ is true of x when either $P_b(x)$ is true—i.e., x is a *b* in the input—or when $P'_b(p(x))$ —i.e., the predecessor of x is a *b* in the *output*. To illustrate, interpreted as part of a logical transduction, this would map an input string *baaa* to *bbbb* as shown below:



To determine the output *bs*, we must identify the set of input positions that satisfy the definition for $P'_b(x)$ in (7). To evaluate this, we notice first that 1 satisfies the first disjunct ($P_b(x)$) of (7). Position 1 thus satisfies $P'_b(x)$. Because 1 satisfies $P'_b(x)$, then, 2 does as well: as its predecessor satisfies $P'_b(x)$, 2 satisfies the second disjunct $P'_b(p(x))$ of (7). Now that we know 2 satisfies $P'_b(x)$, we know that 3 satisfies $P'_b(x)$ for the same reason. In turn, we then also know that 4 satisfies $P'_b(x)$. The function thus labels all of the set $\{1, 2, 3, 4\}$ as *bs* in the output.¹ We can thus model the iterative application of the rule with a least-fixed point logic.

While least-fixed point logics properly extend the power of first-order logic (Libkin, 2004), by restricting ourselves to QF formulae, the functions we define are still *local* in a well-defined sense. The following will then show how this notion of local, recursive logical definitions can capture autosegmental tone mapping patterns. We must first, however, show how to extend the logical perspective to autosegmental representations and the association relation.

2.3 Autosegmental Representations and Association A tone mapping process creates pairs of associated tones and TBUs that are not associated in the input. Consider the one-to-one tone mapping below.

- (9) a. $\begin{array}{c} \mu \mu \mu \dots \\ \vdots \\ \text{HLH} \dots \end{array}$ b. $\begin{array}{c} \mu \mu \mu \dots \\ \vdots \quad \vdots \\ \text{HLH} \dots \end{array}$ c. $\begin{array}{c} \mu \mu \mu \dots \\ \vdots \quad \vdots \quad \vdots \\ \text{HLH} \dots \end{array}$

We may define this mapping process as a logical transduction from an input (no associations) to an output (fully-specified associations). Identifying pairs of output-associated elements is done using a binary association relation $R(x, y)$. The recursion afforded us by least fixed point logic is crucial here, as strings of timing units and melodic units may be arbitrarily long. Implicit definitions like those in (7) allow for precisely this type of iterative association. Below, the definition for the association relation in (9) illustrates, where x and y range over positions on different tiers in the autosegmental representation. Underbraces indicate correspondences between portions of the definition and the mappings they define.

¹ This set that we converge on by recursively interpreting the formula is the *least-fixed point*.

$$(10) \quad R(x, y) \stackrel{d}{=} \underbrace{((\text{first}(x) \wedge \text{first}(y)))}_{(9a)} \vee \underbrace{(R(p(x), p(y)))}_{(9b-c)}$$

This formula has two disjuncts, the first of which defines output association between the first input elements on each tier. In (9a), accordingly, we observe association between the first mora on the timing tier and the first H segment on the melodic tier. The second disjunct is defined implicitly, and states that output association obtains when the predecessors of each domain element are also in the relation. The second mora and first L in (9b) satisfy this, as do the third mora and second H in (9c), etc.

In the next section, we show that these formulas can capture a wide range of tone mapping patterns. This includes directional patterns such as Kikuyu, Mende (Leben, 1973) and Hausa (Newman, 1986, 2000), edge-in association in North Karanga Shona (Odden, 1981; Hewitt & Prince, 1989), and patterns that avoid H-spreading as in Kukuya (Hyman, 1987; Zoll, 2003).

3 Analyses

In the following analyses, we consider logics that assume *either* a function $p(x)$ indicating the immediate predecessor of x or $s(x)$, indicating the immediate successor of x . We also assume predicates $\text{first}(x)$, $\text{penult}(x)$, and $\text{last}(x)$, indicating the first, penultimate, and last member on a tier, respectively. We define $\text{second}(x) \stackrel{d}{=} \text{first}(p(x))$, and $\text{third}(x)$, etc., similarly.

3.1 Mende In Mende, melodies associate to TBUs one to one from left to right (Leben, 1973, 1978):

(11)	a. kó	‘war’	b. pélé	‘house’	c. háwámá	‘waist’
	d. kpà	‘debt’	e. bèlè	‘pants’	f. kpàkàlì	‘three-legged chair’
	g. mbû	‘owl’	h. ngílà	‘dog’	i. fèlámà	‘junction’
	j. mbă	‘rice’	k. nìká	‘cow’	l. ndàvúlá	‘sling’
	m. mbă	‘companion’	n. nyàhâ	‘woman’	o. nìkílì	‘groundnut’

In (i) [fèlámà] ‘junction’, left to right association of the HL melody to TBUs occurs in the following way: H associates to the leftmost mora, L associates to the following mora, and then spreads to the remaining mora.

(12)	a.	$\mu \mu \mu$	b.	$\mu \mu \mu$	c.	$\mu \mu \mu$
		\vdots		$\mid \mid$		$\mid \mid \swarrow$
		HL		HL		HL

Left to right association in Mende is defined $A(x, y) \stackrel{d}{=} R(x, y)$, where $R(x, y)$ is a binary relation defined over variables x (tones) and y (tone bearing units) with three disjuncts. The first disjunct states that the first tone and first TBU associate, as in (12a):

$$(13) \quad \text{first}(x) \wedge \text{first}(y) \quad \mu \mu \mu \quad \mapsto \quad \begin{array}{c} \mu \mu \mu \\ \vdots \\ \text{HL} \end{array}$$

The second disjunct is defined recursively, stating that the association relation R holds between x and y if it also holds between the *predecessors* of x and y . This is observed in (12b):

$$(14) \quad R(p(x), p(y)) \quad \mu \mu \mu \quad \mapsto \quad \begin{array}{c} \mu \mu \mu \\ \mid \mid \\ \text{HL} \end{array}$$

The final disjunct defines association between the last element on the tonal tier x and any element on the TBU tier y for which the predecessor of y is also associated to x , as in (12c). This results in the left to right “spreading” that characterizes Mende tonal association:

$$(15) \text{ last}(x) \wedge R(x, p(y)) \quad \begin{array}{c} \mu \mu \mu \\ | \\ \text{HL} \end{array} \mapsto \begin{array}{c} \mu \mu \mu \\ | \quad \swarrow \\ \text{HL} \end{array}$$

The full definition for tone association in Mende is given in (16).

$$(16) A(x, y) \stackrel{d}{=} R(x, y) \stackrel{d}{=} (\text{first}(x) \wedge \text{first}(y)) \vee (R(p(x), p(y))) \vee (\text{last}(x) \wedge R(x, p(y)))$$

3.2 Kikuyu In Kikuyu, the first tone in a melody associates to the first two TBUs. Then, tone-TBU association proceeds left to right (Clements & Ford, 1979). This is similar to Mende, with the added requirement that the first tone associate to the second TBU as well as the first.

- (17) a. mòtě ‘tree’ b. ìrĩá ‘milk’ c. mòèrékàngéíé ‘releasing self quickly’
 d. ròkǒ! ‘firewood’ e. kàngèrí ‘Kang’eri’ f. kàngèrí ò: níré ‘Kang’eri saw.’
 g. ìkàrá ‘charcoal’ h. rèiðórí! ‘teardrop’ i. ìrĩá né rèèyá ‘Milk is good.’

In (c) [mòèrékàngéíé] ‘releasing self quickly’, an LHLH melody associates to a string of seven TBUs. The first L associates to the first two TBUs. One-to-one, left-to-right association then occurs for the following H, L, and H tone in the melody. Lastly, the final H tone spreads to the final two TBUs.

$$(18) \begin{array}{llll} \text{a. } \begin{array}{c} \mu \mu \mu \mu \mu \mu \mu \\ | \\ \text{L HLH} \end{array} & \text{b. } \begin{array}{c} \mu \mu \mu \mu \mu \mu \mu \\ \swarrow \\ \text{L HLH} \end{array} & \text{c. } \begin{array}{c} \mu \mu \mu \mu \mu \mu \mu \\ \swarrow \quad | \\ \text{L HLH} \end{array} & \text{d. } \begin{array}{c} \mu \mu \mu \mu \mu \mu \mu \\ \swarrow \quad | \quad | \\ \text{L HLH} \end{array} \\ \text{e. } \begin{array}{c} \mu \mu \mu \mu \mu \mu \mu \\ \swarrow \quad | \quad | \quad | \\ \text{L HLH} \end{array} & \text{f. } \begin{array}{c} \mu \mu \mu \mu \mu \mu \mu \\ \swarrow \quad | \quad | \quad \swarrow \\ \text{L HLH} \end{array} & \text{g. } \begin{array}{c} \mu \mu \mu \mu \mu \mu \mu \\ \swarrow \quad | \quad | \quad \swarrow \quad \swarrow \\ \text{L HLH} \end{array} \end{array}$$

Association of tones to TBUs in Kikuyu is defined $A(x, y) \stackrel{d}{=} R(x, y)$, where $R(x, y)$ is a binary relation defined over variables x (tones) and y (tone bearing units) with four disjuncts. The first disjunct states that the first element in the tonal tier and the first element in the TBU tier associate, as in (18a):

$$(19) \text{ first}(x) \wedge \text{first}(y)$$

The second disjunct states that the first element in the tonal tier and the second element in the TBU tier associate, as in (18b).

$$(20) \text{ first}(x) \wedge \text{second}(y)$$

The third disjunct is defined recursively, stating that the association relation R holds between x and y if it also holds between the *predecessors* of x and y . This is observed in (18c,d,e):

$$(21) R(p(x), p(y))$$

The final disjunct defines association between the last element on the tonal tier x and any element on the TBU tier y for which the predecessor of y is also associated to x , as in (18f,g). This results in the left to right “spreading” that accounts for all remaining TBUs in Kikuyu.

$$(22) \text{ last}(x) \wedge R(x, p(y))$$

The full definition for tone association in Kikuyu is given in (23).

$$(23) A(x, y) \stackrel{d}{=} R(x, y) \stackrel{d}{=} (\text{first}(x) \wedge \text{first}(y)) \vee (\text{first}(x) \wedge \text{second}(y)) \vee (R(p(x), p(y))) \vee (\text{last}(x) \wedge R(x, p(y)))$$

3.3 Hausa In Hausa (Newman, 1986, 2000), tones associate one-to-one from right to left, as in the examples below.

- | | | | | | | | | | |
|------|----|--------|----------|----|----------|-------------|----|------------|----------------|
| (24) | a. | jáa | ‘pull’ | b. | jíráa | ‘wait for’ | c. | béebíyáa | ‘deaf mute’ |
| | d. | wàa | ‘who?’ | e. | màcè | ‘woman’ | f. | zàmfarà | ‘Zamfara’ |
| | g. | jàakfí | ‘donkey’ | h. | jìminúu | ‘ostriches’ | i. | bàbbàbbàkú | ‘well roasted’ |
| | j. | fáadi | ‘fall’ | k. | hántúnàa | ‘noses’ | l. | búhúnhúnàa | ‘sacks’ |
| | m. | mântá | ‘forget’ | n. | káràntá | ‘read’ | o. | kákkàràntá | ‘reread’ |

In (h) [jìminúu] ‘ostriches’, association of an LH melody to a timing tier with three morae proceeds as follows: H associates to the rightmost mora, L associates to the penultimate mora, and then to the initial mora.

- | | | | | | | |
|------|----|---------------|----|---------------|----|---------------|
| (25) | a. | $\mu \mu \mu$ | b. | $\mu \mu \mu$ | c. | $\mu \mu \mu$ |
| | | ↓ | | ↓ ↓ | | ↘ ↓ ↓ |
| | | L H | | L H | | L H |

We define right-to-left association in Hausa $A(x, y) \stackrel{d}{=} R(x, y)$, where $R(x, y)$ is a binary relation defined over element variables x (some tone) and y (some mora) and contains three disjuncts. The first drives association between final elements on timing and melodic tiers, as in (25a).

- (26) $\text{last}(x) \wedge \text{last}(y)$

A second disjunct is defined recursively, and states that the relation R obtains between the *successors* of elements x and y . Association in (25b) satisfies this definition.

- (27) $R(s(x), s(y))$

Finally, the third disjunct defines associations for which element x is the first on its tier and relation R holds between that element and the *successor* of element y . This models leftward spreading of a melody-initial tone, as in (25c).

- (28) $\text{first}(x) \wedge R(x, s(y))$

The full definition of right-to-left association in Hausa is given below.

- (29) $A(x, y) \stackrel{d}{=} R(x, y)$
 $R(x, y) \stackrel{d}{=} (\text{last}(x) \wedge \text{last}(y)) \vee (R(s(x), s(y))) \vee (\text{first}(x) \wedge R(x, s(y)))$

3.4 North Karanga Shona Edge-in association is observed in North Karanga Shona non-assertive patterns (Odden, 1981; Hewitt & Prince, 1989). For HLH melodies, first and last tones associate to edges, while the first tone spreads inward up to three TBUs.

- | | | | | |
|------|----|--------------------------|-------------------------------------|---------|
| (30) | a. | ...-p-á | ‘I didn’t give’ | H |
| | b. | ...-tór-à | ‘I didn’t take’ | HL |
| | c. | ...-tór-ès-á | ‘I didn’t make take’ | HLH |
| | d. | ...-tór-és-èr-á | ‘I didn’t make take for’ | HHLH |
| | e. | ...-tór-és-ér-àn-á | ‘I didn’t make take for e.o.’ | HHHLH |
| | f. | ...-tór-és-ér-ès-àn-á | ‘I didn’t make take a lot for e.o.’ | HHHLLH |
| | g. | ...-tór-és-ér-ès-ès-àn-á | (same as f.) | HHHLLLH |

Association of this melody to a timing tier with six TBUs (as in (30f)) is demonstrated below.

- | | | | | | | | | | | |
|------|----|---------------------------|----|---------------------------|----|---------------------------|----|---------------------------|----|---------------------------|
| (31) | a. | $\mu \mu \mu \mu \mu \mu$ | b. | $\mu \mu \mu \mu \mu \mu$ | c. | $\mu \mu \mu \mu \mu \mu$ | d. | $\mu \mu \mu \mu \mu \mu$ | e. | $\mu \mu \mu \mu \mu \mu$ |
| | | ↓ | | ↓ ↘ | | ↓ ↘ ↘ | | ↓ ↘ ↘ ↘ | | ↓ ↘ ↘ ↘ ↘ |
| | | H L H | | H L H | | H L H | | H L H | | H L H |
| | f. | $\mu \mu \mu \mu \mu \mu$ | | | | | | | | |
| | | ↓ ↘ ↘ ↘ ↘ | | | | | | | | |
| | | H L H | | | | | | | | |

We define Shona edge-in association $A(x, y) \stackrel{d}{=} R(x, y)$ where the binary relation $R(x, y)$ ranges over variables representing tones (x) and morae (y) and is defined by five disjuncts. The first disjunct ensures that the first tone and morae associate as in (31a).

$$(32) \quad \text{first}(x) \wedge \text{first}(y) \quad \begin{array}{ccccc} \mu & \mu & \mu & \mu & \mu \\ \text{H} & \text{L} & \text{H} & & \end{array} \mapsto \quad \begin{array}{ccccc} \mu & \mu & \mu & \mu & \mu \\ & & \vdots & & \\ \text{H} & \text{L} & \text{H} & & \end{array}$$

The second disjunct models rightward edge-in spreading of the first tone—via the recursive $R(x, p(y))$ —that does not exceed three TBUs; specifically, it forbids association between that tone and the fourth TBU on the melodic tier or the penult (which will associate to L for a HLH melody). Note that associations in (31b) and (31c) satisfy all four conjuncts.

$$(33) \quad \text{first}(x) \wedge \neg \text{fourth}(y) \wedge \neg \text{penult}(y) \wedge R(x, p(y)) \quad \begin{array}{ccccc} \mu & \mu & \mu & \mu & \mu \\ | & & & & \\ \text{H} & \text{L} & \text{H} & & \end{array} \mapsto \quad \begin{array}{ccccc} \mu & \mu & \mu & \mu & \mu \\ | \swarrow \quad \searrow & & & & \\ \text{H} & \text{L} & \text{H} & & \end{array}$$

Disjuncts three and four model spreading of the melody-internal L elements in a similar way.

$$(34) \quad (\text{second}(x) \wedge \text{fourth}(y) \wedge \neg \text{last}(y)) \vee (\text{second}(x) \wedge \text{third}(y) \wedge \text{penult}(y)) \\ \text{second}(x) \wedge \neg \text{last}(y) \wedge R(x, p(y)) \quad \begin{array}{ccccc} \mu & \mu & \mu & \mu & \mu \\ \swarrow \quad \searrow & & & & \\ \text{H} & \text{L} & \text{H} & & \end{array} \mapsto \quad \begin{array}{ccccc} \mu & \mu & \mu & \mu & \mu \\ \swarrow \quad \searrow \quad \vdots & & & & \\ \text{H} & \text{L} & \text{H} & & \end{array}$$

The former defines association between the second tone (L) and the fourth TBU (if it is not the penult (31d)) or the third (if it is the penult). This accommodates both instances of maximal edge-in spreading of the initial H, and those where it spreads less than three morae. Disjunct four defines rightward spread of the second tone up to the penultimate TBU on the timing tier, as in (31e), again using recursion.

The final disjunct of the definition governs the right edge of the melodic and timing tiers.

$$(35) \quad \text{last}(x) \wedge \text{last}(y) \quad \begin{array}{ccccc} \mu & \mu & \mu & \mu & \mu \\ \swarrow \quad \searrow & & & & \\ \text{H} & \text{L} & \text{H} & & \end{array} \mapsto \quad \begin{array}{ccccc} \mu & \mu & \mu & \mu & \mu \\ \swarrow \quad \searrow \quad \vdots & & & & \\ \text{H} & \text{L} & \text{H} & & \end{array}$$

This drives association between the final elements on each tier, the second half of the edge-in pattern, and is observed in (31f).

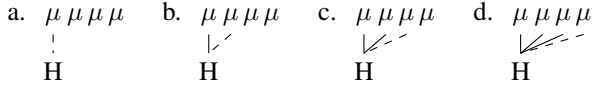
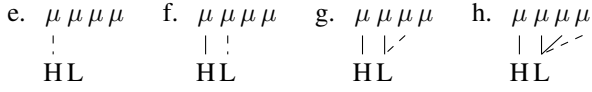
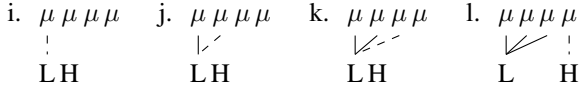
We offer the full definition for association in Shona in (36):

$$(36) \quad A(x, y) \stackrel{d}{=} R(x, y) \\ R(x, y) \stackrel{d}{=} (\text{first}(x) \wedge \text{first}(y)) \vee (\text{first}(x) \wedge \neg \text{fourth}(y) \wedge \neg \text{penult}(y) \wedge R(x, p(y))) \vee \\ ((\text{second}(x) \wedge \text{fourth}(y) \wedge \neg \text{last}(y)) \vee (\text{second}(x) \wedge \text{third}(y) \wedge \text{penult}(y))) \vee \\ (\text{second}(x) \wedge \neg \text{last}(y) \wedge R(x, p(y))) \vee (\text{last}(x) \wedge \text{last}(y))$$

3.5 Kukuya Kukuya exhibits quality-sensitive association (Hyman, 1987). Tones associate left to right, however H spreads only if it is the only tone on the melodic tier. Otherwise, L spreads and H associates to a single TBU.

(37)	a. bá	‘palms’	b. bágá	‘show knives’	c. bálágá	‘fence’
	d. kâ	‘to pick’	e. sámà	‘conversation’	f. kàràgà	‘to be entangled’
	g. sǎ	‘knot’	h. kàrà	‘paralytic’	i. m ^w àrègí	‘younger brother’
	j. bví	‘falls’	k. pàlí	‘goes out’	l. kàlógì	‘turns around’

A logical characterization of association in Kukuya must accommodate a variety of possible melodies. This includes unbounded spreading for melodies with a single H or L tone (a-d below). For HL (e-h) and LH (i-l) melodies, it must restrict association of H to a single TBU and allow L to spread to the remaining TBUs.

(38) *H (or L) melody**HL melody**LH melody*

We define Kukuya quality-sensitive association $A(x, y) \stackrel{d}{=} R(x, y)$ where the binary relation $R(x, y)$ ranges over variables representing tones (x) and morae (y) and is defined by five disjuncts. The first governs association between the first tone in a melody and the first TBU on a timing tier.

(39) $\text{first}(x) \wedge \text{first}(y)$

This disjunct evaluates to true for (38a), (38e), and (38i) above, where initial tones and TBUs associate regardless of quality.

The second, third and fourth disjuncts, by contrast, utilize unary relations which refer to a specific *tone*: P_L for low tone and P_H for high tone. These capture the quality-sensitive nature of spreading in Kukuya. Disjunct two, for example, evaluates to true when association obtains between the second tone and TBU, but only when that second tone is L.

(40) $\text{second}(x) \wedge P_L(x) \wedge \text{second}(y)$

This part of the definition drives association between the second mora and L in HL melodies (38f) while also preventing H association to the second mora in LH melodies (38j). The third disjunct allows L to spread rightward up to the penult (as in (38g), (38j), and (38k)), again using recursion.

(41) $P_L(x) \wedge \neg \text{last}(y) \wedge R(x, p(y))$

Limiting L spread to the penult is necessary in LH melodies, since H associates to the final TBU (38l).

The fourth disjunct targets single H melodies. It allows H to spread rightward—via the recursive $R(x, p(y))$ —but only when it is the only element on the melodic tier (that is, the first and last element):

(42) $P_H(x) \wedge \text{first}(x) \wedge \text{last}(x) \wedge R(x, p(y))$

(38b), (38c), and (38d) are thus well-formed.

The final disjunct associates final elements on respective tiers.

(43) $\text{last}(x) \wedge \text{last}(y)$

This serves two purposes. First, it allows for H to associate to the final TBU in LH melodies (38l). It also permits rightward L spread to reach the *right edge* of a domain when L is the final element on the melodic tier (recall that disjunct three only allows L to spread to the penult). This applies to HL melodies (as in (38h)) as well as L melodies.

The full definition for Kukuya association is given below.

(44) $A(x, y) \stackrel{d}{=} R(x, y)$
 $R(x, y) \stackrel{d}{=} (\text{first}(x) \wedge \text{first}(y)) \vee (\text{second}(x) \wedge P_L(x) \wedge \text{second}(y)) \vee$
 $(P_L(x) \wedge \neg \text{last}(y) \wedge R(x, p(y))) \vee (P_H(x) \wedge \text{first}(x) \wedge \text{last}(x) \wedge R(x, p(y))) \vee$
 $(\text{last}(x) \wedge \text{last}(y))$

4 Discussion

This paper presents a characterization of tone-to-TBU association that is quantifier-free and uses least-fixed point operators. This shows that tone association patterns can be described with *local* functions, a level of computational complexity that aligns with restrictions on phonology in general. Additionally, the combination of QF-definable functions with the recursion afforded by least-fixed point logic describes processes that are *output-oriented*.

The hypothesis that tone mapping patterns are least-fixed point, quantifier free functions compares well to other approaches. In derivational frameworks, for example, association conventions are stipulated as left-to-right, right-to-left, or edge-in (Leben, 1973; Goldsmith, 1976; Hewitt & Prince, 1989)—there is no explanation why a ‘center-outward’ convention is not possible. In OT treatments (Yip, 2002; Zoll, 2003), there is no explanation for why some ALIGN constraints are in CON and not others, such as the logically possible ALIGN constraints that can produce centering patterns (Eisner, 1997). Furthermore, standard OT treatments cannot capture N. Karanga Shona (Zoll, 2003). In contrast, the least fixed point characterization put forth here describes a wide range of patterns, including those which earlier work has failed to capture. It also explains why centering patterns are *not* attested: they cannot be defined through local structures in the output.

Earlier, we state that formulas defining association over autosegmental representations refer to either the predecessor or successor of some domain element, but never both. This restriction is necessary to avoid predicting unattested association patterns such as those exhibiting line crossing. Consider, for example, a formula containing least fix point operators which call p and s functions:

$$(45) \quad A(x, y) \stackrel{d}{=} R(x, y) \\ R(x, y) \stackrel{d}{=} (\text{first}(x) \wedge \text{first}(y)) \vee (\text{last}(x) \wedge \text{last}(y)) \vee \\ (R(x, p(y))) \vee (R(p(x), p(y))) \vee (R(x, s(y))) \vee (R(s(x), s(y)))$$

This formula establishes association between the first elements on both tiers as well as the final elements. Then, using reference to both predecessor and successor in recursive definitions, it permits domain-wide leftward and rightward spreading. Such a definition would allow for massively crossing association lines, as below.

$$(46) \quad \begin{array}{c} \mu \mu \mu \mu \\ \diagdown \diagup \diagdown \diagup \\ TTTT \end{array}$$

So far, this restriction is stipulative, but future work may be able to derive it formally. One possibility is that reference to predecessor or successor in these formulae corresponds to the non-intersecting space of left and right output strictly local functions in the sub-regular hierarchy (Heinz, 2010; Chandlee, 2014). These are precisely the local, output-oriented class of functions we use to define tonal association. Previous work has given automata-theoretic characterizations of these functions (Chandlee, 2014), but their expression in logical terms has yet to be explored. Proving this equivalency is beyond the scope of the current study, but future work may continue to explore the relationship between representations of phonological processes using logical and automata-theoretic characterizations.

5 Conclusion

This paper articulated a computational theory of tone capable of distinguishing between tone mapping patterns that are possible and those that are impossible using a restricted *least fixed point logic* (Libkin, 2004). The benefit of a computational characterization of tone mapping is that it provides a restrictive, testable, and learnable theory of the tone mapping function itself. It provides a characterization of tone mapping that is independent of formal linguistic theory and so applicable to the study of tone no matter the particular linguistic formalism adopted.

A number of attested tone mapping patterns were captured, including left-to-right and right-to-left patterns, such as in Kikuyu, Mende (Leben, 1973) and Hausa (Newman, 1986, 2000); edge-in patterns such as in North Karanga Shona (Odden, 1981; Hewitt & Prince, 1989); and quality-sensitive patterns such as in Kukuya (Hyman, 1987; Zoll, 2003). Unattested patterns such as those in (1b) are excluded.

Importantly, the least fixed point typology of tone mapping exceeds that of both derivation-based frameworks (Zoll, 2003) and optimization-based frameworks (Prince & Smolensky, 1993) in terms of empirical coverage. It addresses an issue raised in (Jardine, 2017a), that found that simple one-to-one mapping of tones to TBUs as a function lies beyond established boundaries on phonological complexity. It also establishes a computational definition of output locality in non-linear structures.

References

- Chandlee, Jane (2014). *Strictly Local Phonological Processes*. Ph.D. thesis, University of Delaware.
- Chandlee, Jane & Jeffrey Heinz (2018). Strict locality and phonological maps. *LI* 49, 23–60.
- Chandlee, Jane & Steven Lindell (forthcoming). A logical characterization of strictly local functions. Heinz, Jeffrey (ed.), *Doing Computational Phonology*, OUP.
- Clements, George N. & Kevin C. Ford (1979). Kikuyu tone shift and its synchronic consequences. *Linguistic Inquiry* 10, 179–210.
- Courcelle, Bruno (1994). Monadic second-order definable graph transductions: a survey. *Theoretical Computer Science* 126, 53–75.
- Eisner, Jason (1997). What constraints should OT allow? Talk handout, Linguistic Society of America, Chicago. ROA#204-0797. Available at <http://roa.rutgers.edu/>.
- Engelfriet, Joost & Hendrik Jan Hoogeboom (2001). MSO definable string transductions and two-way finite-state transducers. *ACM Transactions on Computational Logic* 2, 216–254.
- Goldsmith, John (1976). *Autosegmental Phonology*. Ph.D. thesis, Massachusetts Institute of Technology.
- Heinz, Jeffrey (2010). Learning long-distance phonotactics. *LI* 41, 623–661.
- Heinz, Jeffrey (2018). The computational nature of phonological generalizations. Hyman, Larry & Frans Plank (eds.), *Phonological Typology, Phonetics and Phonology*, De Gruyter Mouton, chap. 5, 126–195.
- Hewitt, Mark & Alan Prince (1989). OCP, locality, and linking: the N. Karanga verb. Fee, E. Jane & Katherine Hunt (eds.), *WCCFL* 8, Stanford: CSLI Publications, 176–191.
- Hyman, Larry (1987). Prosodic domains in Kikuyu. *NLLT* 5, 311–333.
- Jardine, Adam (2017a). The local nature of tone-association patterns. *Phonology* 34, 385–405.
- Jardine, Adam (2017b). On the logical complexity of autosegmental representations. Kanazawa, Makoto, Philippe de Groote & Mehrnoosh Sadrzadeh (eds.), *Proceedings of the 15th Meeting on the Mathematics of Language*, Association for Computational Linguistics, London, UK, 22–35.
- Leben, W. R. (1973). *Suprasegmental phonology*. Ph.D. thesis, Massachusetts Institute of Technology.
- Leben, W. R. (1978). The representation of tone. Fromkin, Victoria (ed.), *Tone—A Linguistic Survey*, Academic Press, 177–219.
- Libkin, Leonid (2004). *Elements of Finite Model Theory*. Berlin: Springer-Verlag.
- Newman, Paul (1986). Tone and affixation in Hausa. *Studies in African Linguistics* 17:3, 249–267.
- Newman, Paul (2000). *The Hausa Language: An encyclopedic reference grammar*. New Haven: Yale University Press.
- Odden, David (1981). *Problems in tone assignment in Shona*. Ph.D. thesis, University of Illinois, Urbana-Champaign.
- Prince, Alan & Paul Smolensky (1993). Optimality Theory: Constraint interaction in generative grammar. *Rutgers University Center for Cognitive Science Technical Report* 2.
- Rogers, James (1997). Strict It2 : Regular :: Local : Recognizable. Retoré, Christian (ed.), *Logical Aspects of Computational Linguistics: First International Conference, LACL '96 Nancy, France, September 23–25, 1996 Selected Papers*, Springer Berlin Heidelberg, Berlin, Heidelberg, 366–385.
- Williams, Edwin S. (1976). Underlying tone in Margi and Igbo. *Linguistic Inquiry* 7:3, 463–484.
- Yip, Moira (2002). *Tone*. Cambridge University Press.
- Zoll, Cheryl (2003). Optimal tone mapping. *LI* 34:2, 225–268.